

REMARKS

The claims remaining in the present application are Claims 2-11, 13, 15, 17-20, 22, 24, 25-27, and 29-37. Claims 2-4, 7-11, 13, 15, 19-20, 22, 24 and 26 have been amended. Claims 1, 12, 14, 16, 21, 23 and 28 have been cancelled, without prejudice. Claims 29-37 have been added. No new matter has been added as a result of these claim amendments.

35 U.S.C. §112

The rejection asserts that, “translating from PERL to an intermediate language and translating from said intermediate language to the format that is suitable for the CORBA” is not disclosed in the specification in such a way as to convey to those of ordinary skill in the art that the inventors had possession of the claimed invention at the time the application was filed. The Applicants respectfully traverse this rejection. The specification discloses the claimed limitations in several locations. Applicants refer to Figure 6 and associated text. However, the rejection is moot in light of the claim amendment. However, Applicants reserve the right to re-enter these claim limitations.

35 U.S.C. §103

Claims 3, 9, 14-15, 18 and 26-27 are rejected under 35 U.S.C. §103(a) as being unpatentable over Nevarez in view of Goldberg, U.S. Patent No. 6,496,833 (hereinafter, Goldberg). Claim 14 has been cancelled, without

prejudice. Therefore, the rejection to Claim 14 is moot. The rejection to Claims 3, 9, 15, 18 and 26-27 is respectfully traversed, for the following reasons.

Amended Independent Claim 3 recites:

A method for allowing a single process executing in a processor and comprising a Practical Extraction Report Language (PERL) program to directly access a distributed object via a Common Object Request Broker Architecture (CORBA), comprising:

- a) accessing a request from a PERL program code specifying said distributed object;
- b) an adapter program code of said process translating said request to a format that is substantially compliant with a client stub program of the process;
- c) said client stub program code making a call to access said distributed object via a CORBA;
- d) said client stub program code receiving a response from said call;
- e) translating said response to a form which is substantially compliant with a Practical Extraction Report Language; and
- f) passing said translated response to said PERL program code, wherein said single process comprising the PERL program code directly accesses said distributed object.

Support for amendments to Claim 3 may be found in the specification at least at page 9, lines 1-9, and page 24, line 2 - page 26, line 47.

The Applicants have claimed a method in which a single process executing in a processor and comprising a PERL program to directly access a distributed object via CORBA. The Applicants have claimed translating a request from a PERL program code to a format that is substantially compliant with a client stub in the process executing in the processor. This enables the executing process comprising the PERL program code to make calls directly to CORBA objects. In contrast, Nevarez teaches a universal component system product 224, which helps a PERL program assess certain objects, provided they are registered with the universal component system product.

Because the Applicants have claimed a method that allows PERL to effectively have direct access to the distributed object, there is no need for a separately executing software process accesses the distributed object. In Nevarez system, the universal component system product 224 executes separately from the PERL program. If the universal component system product 224 is not available for some reason, the PERL program will not be able to access the distributed object.

Applicants have claimed an adapter program code executing in the same process with said PERL program code translates a call from a PERL program. Applicants respectfully assert that Nevaraz fails to teach or suggest an adapter program code executing in the same process with said

PERL program code, as claimed. In contrast, Navarez teaches a mapping by the universal component system product 224, which executes as a separate process from the PERL program.

Nevarez teaches a USC core implementation that serves as a separately executing bridge process between registered programming languages and registered object libraries. The bridge process requires what Nevarez refers to as “component connection aids.” For example, the programming language must have a template registered (col. 11, line 36 - col. 12, line 20). Further, the object library must have information about the object model registered (col. 12, lines 21-40). Significantly, Nevarez teaches that the PERL program is dependent upon the UCS component. For instance, if the UCS component is it is not available, then the PERL program code cannot access the distributed object.

Further, Applicants do not understand Nevarez to explicitly teach a conversion from PERL to CORBA. Applicants appreciate that Nevarez is providing an example in Figure 2 and associated text. However, Applicants respectfully assert that the example fails to disclose a way for a PERL compliant program to communicate with a CORBA compliant object. For example, Figure 2 of Nevarez fails to depict CORBA. At col. 4 line 66 - col. 5 line 4, Nevarez asserts that Java beans and COM objects can be adapted by

the object model adapter. CORBA objects are conspicuously absent from this list.

When Nevarez does mention CORBA, Applicants note that the discussion is generalized and does not specifically assert a way for a PERL program to communicate with a CORBA compliant object. For example, at col. 7, line 25 - col. 8, line 2, Nevaraz provides a general background discussion about objects and object models. Navarez indicates that CORBA is one type of object model, but does not assert in this passage any conversions between PERL and CORBA. Thus, Applicants respectfully assert that Navarez does not teach the claimed

With respect to the rejection's reference of col. 7 lines 40-45 of Nevarez, this passage provides background information about methods. However, this passage does not provide any teaching related to the claimed limitation of translating a call from said PERL program to a format substantially compliant with a client stub program.

Goldberg does not remedy the deficiency in Navarez with respect to the discussed limitations in that even if Goldberg's teaching of a client stub were to be combined with Nevarez' teaching, the Applicants claimed invention would not be arrived at. Applicants have claimed that the client stub (which accesses the CORBA object) is executing in the same process as the PERL program. Navarez specifically teaches that the PERL program is a

separate process from the universal component system product 224.

Navarez is silent as to the use of a client stub program. However, to the Applicants understanding of Navarez, if a client stub were to be integrated into Navarez' system it would be within the universal component system product 228, which executes as a separate process from the PERL program.

For the foregoing reasons, Nevarez and Goldberg fail to teach or suggest the limitations of Claim 3, alone or in combination. Therefore, Applicants respectfully request allowance of Claim 3.

Dependent Claims 9, 15, 18 and 26-27 incorporate similar limitations as the limitations discussed in the response to Claim 3. For at least the reasons discussed in the response to Claim 3, Claims 9, 15, 18 and 26-27 are respectfully believed to be allowable.

CLAIM 26

Claim 26 is respectfully believed to be allowable for the following additional reasons. Claim 26 recites limitations of the adapter program handling an exception that relates to said PERL program. Applicants respectfully assert that Nevarez and Goldberg fail to teach or suggest the limitations of Claim 26, alone or in combination.

Goldberg at col. 7, lines 58-59 may disclose that "if an error has occurred, exception information generated by the server or by the ORB is

returned. Applicants understand that the exception will be handled by some code. However, Applicants respectfully assert that Goldberg does not teach that an adapter handles an error related to the PERL program.

Applicants respectfully assert that Nevarez does not remedy the deficiency in Goldberg with respect to the discussed limitations in that Nevarez fails to teach or reasonably suggest an adapter handling an error related to the PERL program.

For the foregoing reasons, Claim 26 is respectfully believed to be patentable over the prior art.

35 U.S.C. §102

Claims 1-2, 4-8, 10-13, 16-17, 19-25 and 28 stand rejected under 35 U.S.C. §102 as being anticipated by Nevarez et al. U.S. Patent 6,609,158 (hereinafter, Nevarez). Claims 1, 12, 16, 21, 23, and 28 have been cancelled, without prejudice. Therefore, the rejection to Claims 1, 12, 16, 21, 23, and 28 is moot. The rejection to Claims 2, 4-8, 10-11, 13, 17, 19-20, 22 and 24-25 is respectfully traversed for the reasons below.

Claims 11 and 20 recite similar limitations to those discussed in the response to Claim 3 under the heading for the rejection under 35 U.S.C. §103. For reasons discussed in the response to Claim 3, Nevarez fails to

teach or suggest the limitations of Claims 11 or 20. As such, Applicants respectfully request the allowance of Claims 11 and 20.

Claims 2, 4-8, 10, 13, 17, 19, 22 and 24-25 depend from Claims 3, 11, and 20 and are respectfully believed to derive patentability at least therefrom.

CLAIM 2

Claim 2 is respectfully believed to be allowable for the following additional reasons. Claim 2 recites limitations of the adapter program converting a data structure specified by said PERL request into a form that is substantially compliant with the client stub. In contrast, to converting a data structure, Nevarez teaches a universal language adapter places a script command into a form that is recognized by the UCS core. For example, Nevarez teaches that commands are placed into calls to create or delete objects or to invoke methods. However, Nevarez is silent as to the claimed converting a data structure specified by said PERL request into a form that is substantially compliant with said client stub.

Because Nevarez fails to teach or suggest the claimed converting a data structure, Claim 2 is respectfully believed to be patentable.

NEW CLAIMS

Claims 29-37 have been added. Support for Claims 29-30 may be found in the specification at least at page 24, line 40, page 25, line 48. Support for Claims 31-34 may be may be found in the specification at least at page 9 line 5 - page 10 line 10. Support for new Claims 35-37 may be found in the specification at least at page 14, lines 1-6.

Independent Claim 31 recites in part:

wherein said PERL application program, said client stub, and said adaptor module execute in a single process on said processor, and wherein said process is able to directly access said distributed object.

Claim 31 is respectfully believed to be patentable at least because the prior art fails to teach or reasonably suggest, "said PERL application program, said client stub, and said adaptor module execute in a single process on said processor."

Further, Claim 31 is respectfully believed to be allowable because the prior art fails to teach or reasonable suggest, "said process is able to directly access said distributed object." For example, Nevarez teaches that an intermediary must be used between the PERL program and the distributed object. The intermediary (e.g., UCS product 224) causes a delay and also creates an undesired dependency.

Claims 32-34 are respectfully believed to derive patentability at least from their dependency from Claim 31.

Claims 33-35 are respectfully believed to derive patentability at least from their dependency from Claim 3.

Claims 29-30 are respectfully believed to derive patentability at least from their dependency from Claim 24.

CONCLUSION

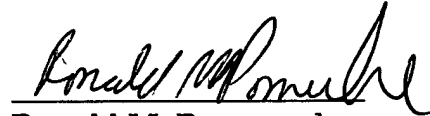
Based on the amendments and arguments presented above, it is respectfully submitted that Claims 2-11, 13, 15, 17-20, 22, 24, 25-27, and 29-37 overcome the rejections of record. Therefore, allowance of these Claims is respectfully solicited.

The Applicants respectfully request that the Examiner contact the Applicants' undersigned representative at the below listed telephone number so that an Examiner interview may be scheduled regarding the instant amendment and response.

Respectfully submitted,

WAGNER, MURABITO & HAO LLP

Dated: 12/7, 2004



Ronald M. Pomeranke
Registration No. 43,009

Address:

WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone:

(408) 938-9060 Voice
(408) 938-9069 Facsimile